

FPGA Console

硬件实现的 VT220 兼容终端

计63 陈晟祺 2016010981

计64 周聿浩 2016011347

<https://github.com/Harry-Chen/fpga-virtual-console>

项目背景

What is...

- *Console*: 控制台，一种 I/O 设备，自身无处理能力
- *Virtual Console*: 操作系统使用软件模拟的控制台
- *Terminal Emulator*: `cmd.exe`, `gnome-terminal`, `screen`, etc.,
- *VT220*: 通信规范，规定计算机与终端如何双向通信
- *XTERM-256color*: 更现代的规范，顾名思义有颜色支持

What we have done:

- 使用 Cyclone IV EP4CE55F23I7 实现的物理终端
- 串口连接计算机，PS/2 连接键盘，HDMI 连接显示器
- 实现 VT220 与 XTERM-256color 中与颜色、光标有关部分

Speak English!

- 能开 Vim 和 Nano (Emacs是异端)
- Talk is cheap, show me the demo

项目特性

高分辨率

标准：640*480@60Hz VGA 输出，80列*25行字符
实现：800*600@72Hz VGA 输出，100列*50行字符

高速度

命令解析、图形渲染全流水线工作，时许允许最高频率 120 MHz
串口波特率高达 3M，能以 25 FPS 播放全彩电影（字符模式）

高稳定性

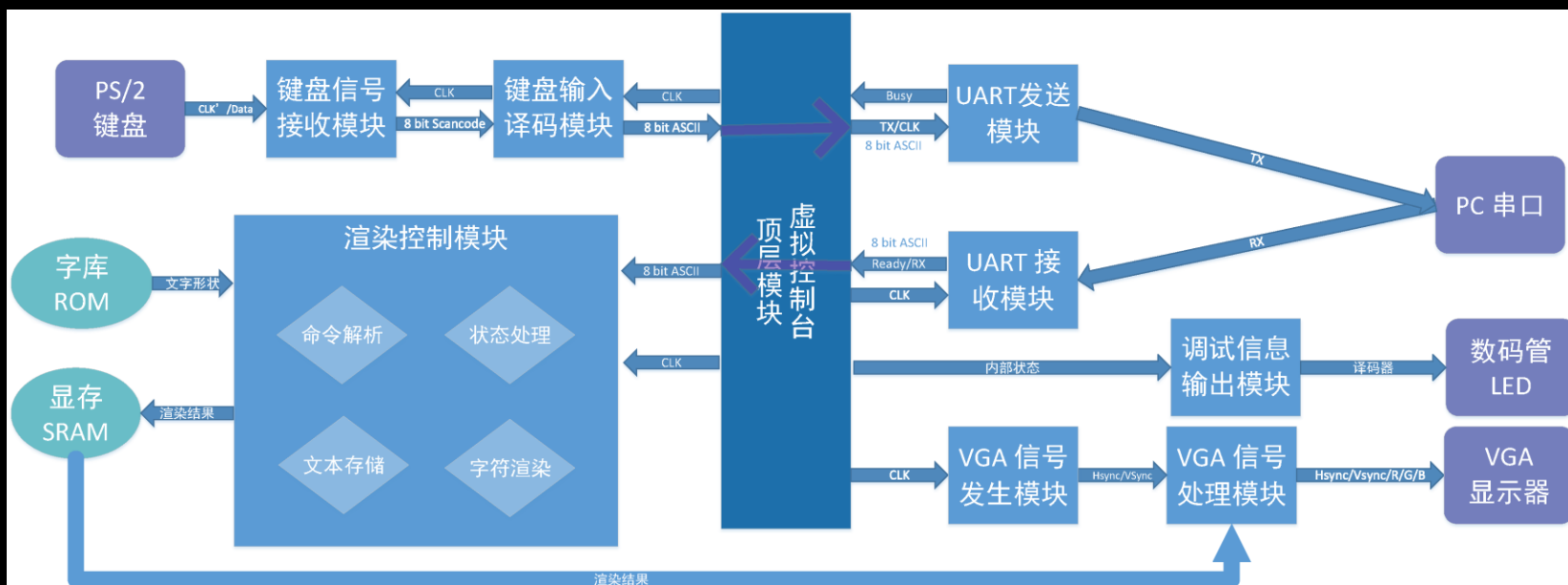
使用两块显存双缓冲（*Double Buffering*），防止撕裂、抖动
充分利用 SRAM，读写稳定运行于 50MHz（注：非原实验板）

高模块化

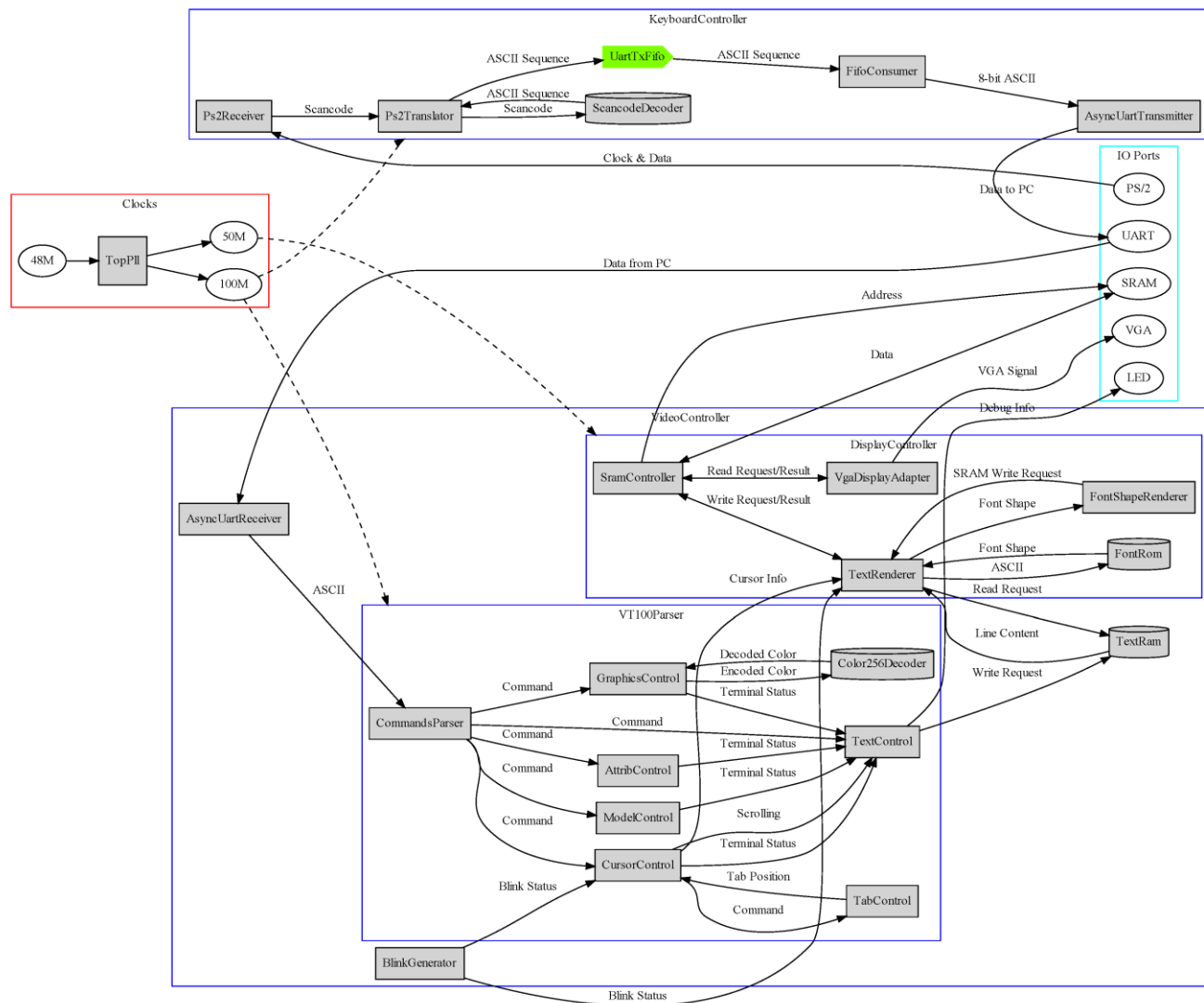
纯 SystemVerilog 实现，代码风格一致、可读性强
各部分独立工作，高内聚、低耦合，I/O 模块具有通用性

Demonstration

项目架构：设计

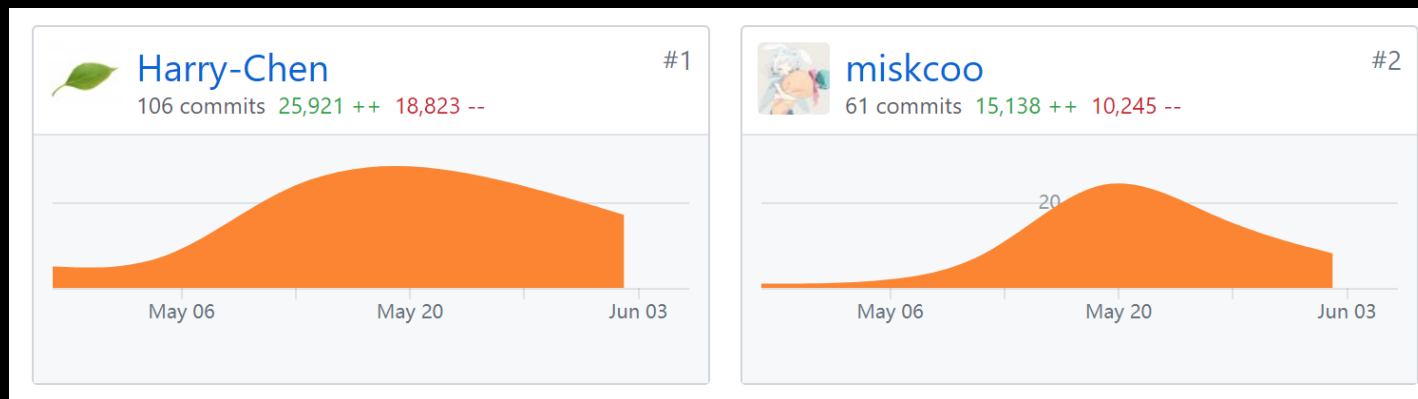


项目架构：实现



项目分工

"All programmers are created equal."

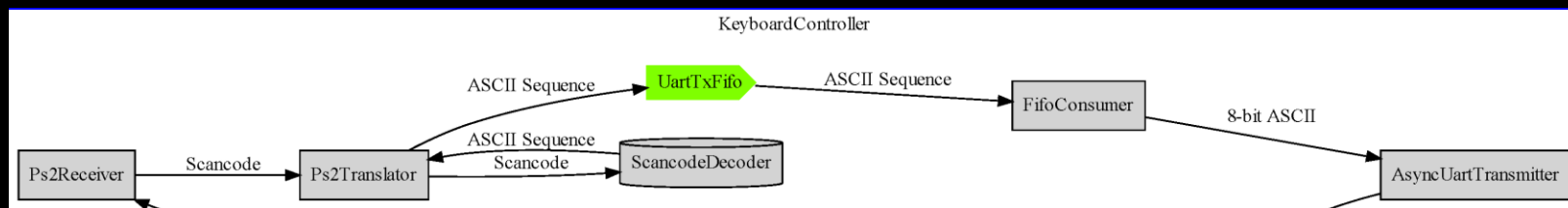
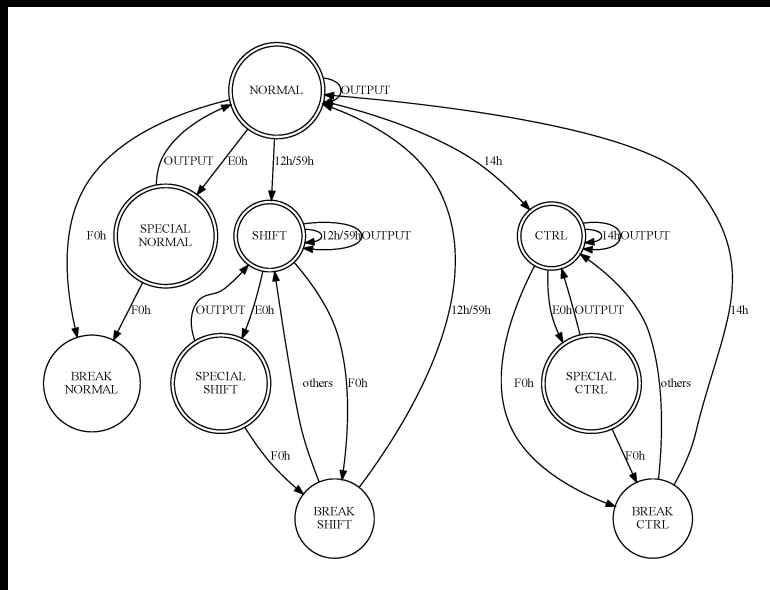


实现细节：键盘

- 数据流向：
 - PS/2→扫描码→Escape Sequence→FIFO→串口

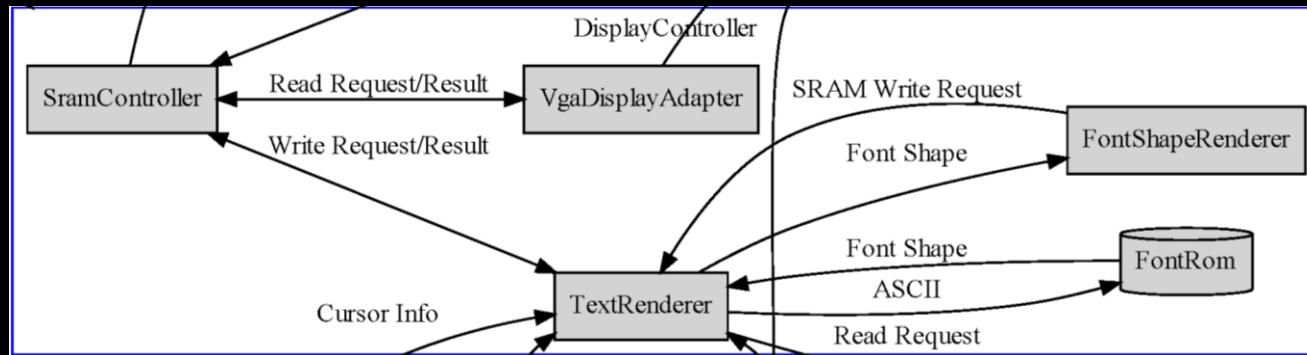
- 特殊处理：
 - Shift, Ctrl, Shift + Ctrl
 - 长扫描码 (E0 xx)
 - 长转义序列 (ESC [1;5A)

- Why FIFO?
 - 波特率 < 产生字符的速率
 - 模块解耦合



实现细节：渲染与 VGA

- SRAM 控制器：
 - 最长读/写时间在 20 ns 左右，最高允许 50 MHz
 - 接受请求类型：
 - VGA 读，像素频率 50 MHz，不允许读取失败
 - 渲染器写，状态机实现，可以进行等待
 - 两者非严格的轮流关系（消隐区、渲染空闲时间片）
- 最终实现：
 - 每行存储相邻两个像素，方便寻址
 - 优先处理 VGA 读请求，读空闲时进行写入

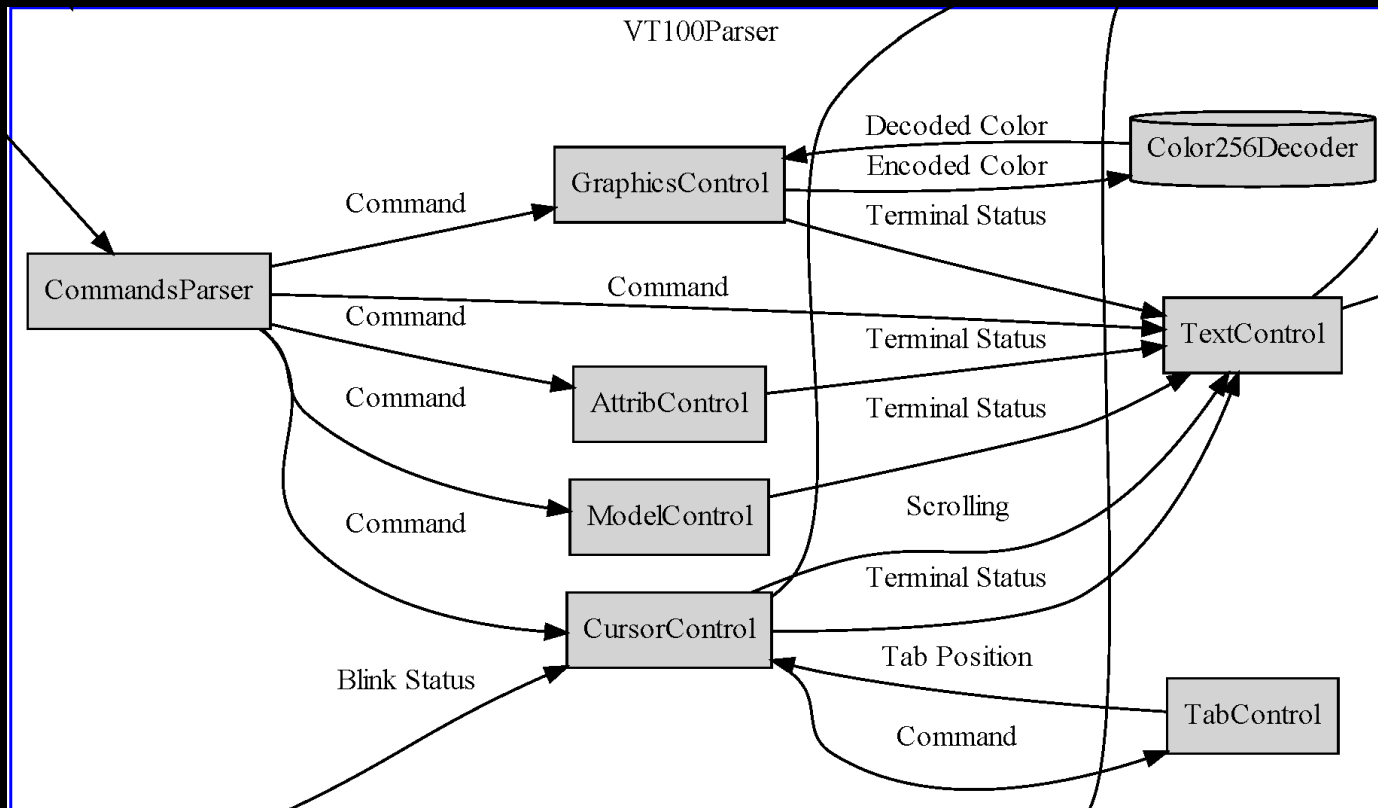


实现细节：命令解析(1)

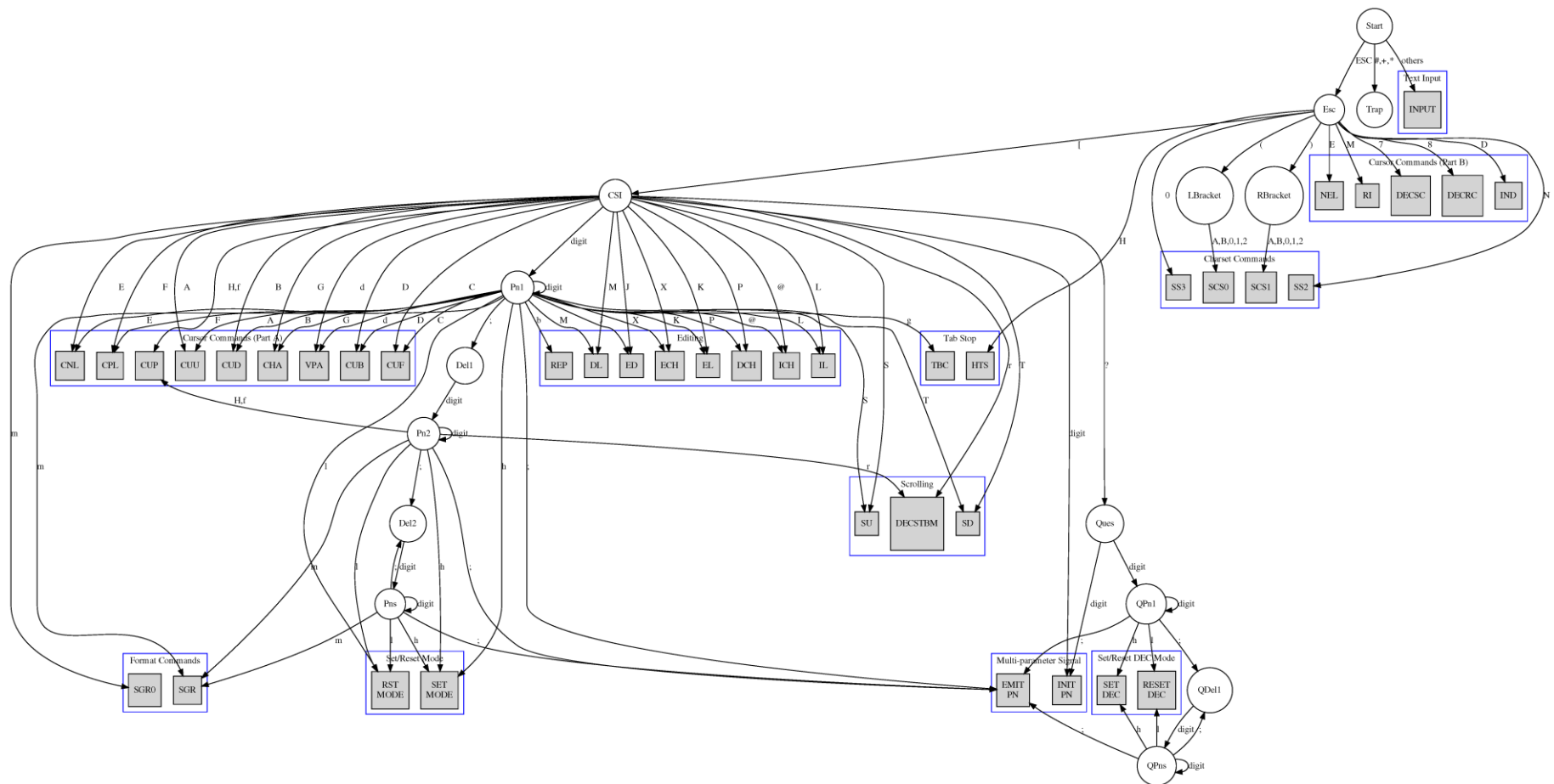
- 命令是长成什么样？
 - ESC[48;2;255;255;0;39;2m 设置背景色、前景色、特效
 - ESC[?25;12h 光标可见并且闪烁
 - ESC[100b 重复输入最后一次输入的字符100次
-
- 命令大约有30多条
 - 大部分带有参数，少数参数个数不定（例如颜色设置）

实现细节：命令解析(2)

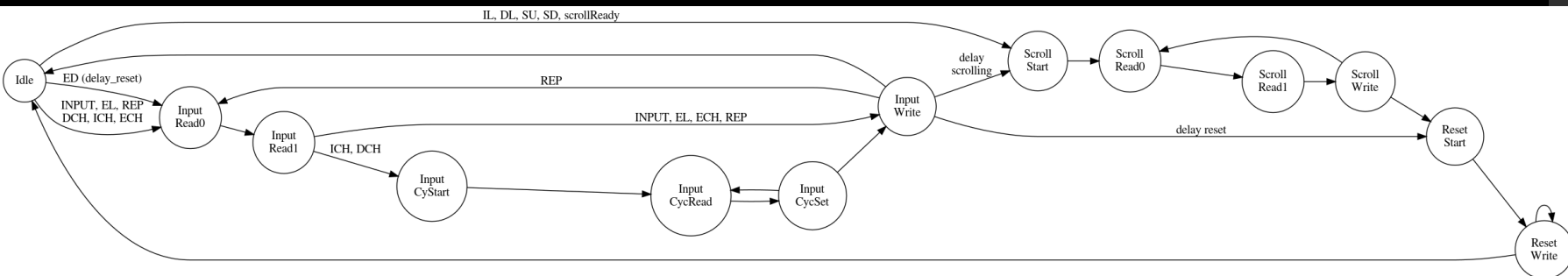
- You don't want to know the details, do you?



- ## 实现细节：命令解析(3)



实现细节：文本编辑



1. 行内编辑

主要负责对一行内的信息进行编辑，一共分为Set、Insert、Remove三种编辑方法

2. 屏幕滚动

主要负责将某区域上移/下移指定行数。

3. 多行清空

主要负责将某区域填充为空字符。

Questions